

Performance of Floor Acquisition Multiple Access in Ad-Hoc Networks

J.J. Garcia-Luna-Aceves[†] and Chane L. Fullmer[‡]

Abstract— The performance of the FAMA-NCS protocol in ad-hoc networks is analyzed. FAMA-NCS (for floor acquisition multiple access with non-persistent carrier sensing) guarantees that a single sender is able to send data packets free of collisions to a given receiver at any given time. FAMA-NCS is based on a three-way handshake between sender and receiver in which the sender uses non-persistent carrier sensing to transmit a request-to-send (RTS) and the receiver sends a clear-to-send (CTS) that lasts much longer than the RTS to serve as a “busy tone” that forces all hidden nodes to back off long enough to allow a collision-free data packet to arrive at the receiver. It is shown that that FAMA-NCS performs better than ALOHA, CSMA, and all prior proposals based on collision avoidance dialogues (e.g., MACA, MACAW, and IEEE 802.11 DFWMAC) in the presence of hidden terminals. Simulations experiments are used to confirm the analytical results.

I. INTRODUCTION

The medium access control (MAC) protocol with which packet-radios (or stations) can share a common broadcast channel is essential in a packet-radio network. CSMA (carrier sense multiple access) protocols [9] have been used in a number of packet-radio networks in the past [10]; these protocols attempt to prevent a station from transmitting simultaneously with other stations within its transmitting range by requiring each station to listen to the channel before transmitting.

The hardware characteristics of packet-radios are such that a packet-radio cannot transmit and listen to the same channel simultaneously; therefore, collision detection (CSMA/CD [12]) cannot be used in a single-channel packet-radio network. The throughput of CSMA protocols is very good, as long as the multiple transmitters within range of the same receivers can sense one another’s transmissions. Unfortunately, “hidden terminal” problems [17] degrade the performance of CSMA substantially, because carrier sensing cannot prevent collisions in that case.

The busy tone multiple access (BTMA) protocol [17] was the first proposal to combat the hidden-terminal problems of CSMA. BTMA is designed for station-based networks and divides the channel into a message channel and the busy-tone channel. The base station transmits a busy-tone signal on the busy-tone channel as long as it senses carrier on the data channel. Because the base station is in line of sight of all terminals, each terminal can sense the busy-tone channel to determine the state of the data channel. The limitations of BTMA are the use a separate channel to convey the state of the data channel, the need for the receiver to transmit the busy tone while detecting carrier in the data channel, and the difficulty of detecting the busy-tone signal in a narrowband channel.

A receiver initiated busy-tone multiple access protocol for packet-radio networks has also been proposed [22]. In this scheme, the sender transmits a request-to-send (RTS) to the receiver, before sending a data packet. When the receiver obtains a correct RTS, it transmits a busy tone in a separate channel to alert other sources nearby that they should backoff. The correct source is always notified that it can proceed with transmission of the data packet. The limitations of this scheme is that it still requires a separate busy-tone channel and full-duplex operation at the receiver.

One of the first protocols for wireless networks based on a handshake between sender and receiver was the SRMA (split-channel

reservation multiple access) [18]. According to SRMA, the sender of a packet uses ALOHA or CSMA to decide when to send a request-to-send (RTS) to the receiver. In turn, the receiver responds with a clear-to-send (CTS) if it receives the RTS correctly; the RTS tells the sender when to transmit its data packet. Although SRMA was proposed with one or two control channel for the RTS/CTS exchange, the same scheme applies for a single channel.

Since the time SRMA was first proposed, several other medium access control (MAC) protocols have been proposed for either single-channel wireless networks or wireline local area networks that are based on similar RTS-CTS exchanges, or based on RTSs followed by pauses [1], [20], [4], [11], [13], [15]. Karn [8] proposed a protocol called MACA (multiple access collision avoidance) to address the problems of hidden terminals in single-channel networks. MACA amounts to a single-channel SRMA using ALOHA for the transmission of RTSs; it attempts to detect collisions at the receiver by means of the RTS-CTS exchange without carrier sensing. The IEEE 802.11 MAC protocol for wireless LANs includes a transmission mode based on an RTS-CTS handshake (DFWMAC [3], [7]).

In this paper, we analyze a new variation on MAC protocols based on RTS-CTS exchanges that is particularly attractive for ad-hoc networks. We call this protocol FAMA-NCS (floor acquisition multiple access with non-persistent carrier sensing). The objective of FAMA-NCS is for a station that has data to send to acquire control of the channel in the vicinity of the receiver (which we call “the floor”) before sending any data packet, and to ensure that no data packet collides with any other packet at the receiver.

Ensuring that floor acquisition is enforced among competing senders hidden from one another and who have requested the floor (i.e., sent an RTS) can only be achieved by the receivers. Accordingly, in FAMA-NCS, the length of a CTS is longer than the duration of an RTS and ensures that the CTS from a receiver lasts long enough for any hidden sender that did not hear the RTS being acknowledged to hear what amounts to a jamming signal from the receiver. Section II describes FAMA-NCS, which was first introduced in [6]. Section III analyzes the throughput of FAMA-NCS in ad-hoc networks using an approximate model of the way in which nodes are connected. Section IV compares by simulation the performance of FAMA-NCS with MACAW and DFWMAC. Our results show very clearly that carrier sensing at the sender and the longer duration of CTSs compared to RTSs are critical to the performance and simplicity of MAC protocols based on RTS-CTS handshakes for ad-hoc networks in which nodes can transmit packets asynchronously. The simulations also help to validate our analytical results. FAMA-NCS is the first proposal based on RTS-CTS exchanges that recognizes the importance of making a CTS act as a busy tone for half-duplex operation.

II. FAMA PROTOCOL

A. Overview

FAMA-NCS requires a station who wishes to send one or more packets to acquire the floor before transmitting the packet train. The floor is acquired using an RTS-CTS exchange multiplexed together with the data packets in such a way that, although multiple RTSs and CTSs may collide, data packets are always sent free of collisions. The basic principles of floor acquisition are inspired on earlier work by Kleinrock and Tobagi on BTMA [17], the use of

This work was supported in part by the Defense Advanced Research Projects Agency under Grants DAAB07-95-C-D157 and DAAH04-96-1-0210.

[†]J.J. Garcia-Luna-Aceves is with the Computer Engineering Department, School of Engineering, University of California, Santa Cruz, CA 95064, USA. (email: jj@cse.ucsc.edu)

[‡]Chane L. Fullmer is with Rooftop Communications Corporation, Mt. View, CA. 94041, USA. (email: chane@rooftop.com)

Report Documentation Page			Form Approved OMB No. 0704-0188		
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>					
1. REPORT DATE 1998	2. REPORT TYPE	3. DATES COVERED 00-00-1998 to 00-00-1998			
4. TITLE AND SUBTITLE Performance of Floor Acquisition Multiple Access in Ad-Hoc Networks			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of California at Santa Cruz, Department of Computer Engineering, Santa Cruz, CA, 95064			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF: a. REPORT unclassified			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 6	19a. NAME OF RESPONSIBLE PERSON
b. ABSTRACT unclassified					
c. THIS PAGE unclassified					

RTS-CTS exchanges first described for SRMA [18], and the provision of priorities among packets introduced for the transmission of priority acknowledgments in ALOHA and CSMA [19].

To acquire the floor, a station sends an RTS using either packet sensing or carrier sensing. The first variant corresponds to using the ALOHA protocol for the transmission of RTSs; the second consists of using a CSMA protocol to transmit RTSs. A station sends a CTS after receiving an error-free RTS addressed to it. When a station receives an error-free CTS, it knows that the floor has been acquired by the station to whom the CTS is addressed. After floor acquisition the floor holder is able to send data packets free of collisions over the channel. Any reliable link control scheme can be implemented on top of FAMA-NCS between the floor holder and the stations with whom it wishes to communicate. This is accomplished by forcing stations that do not have the floor to wait a predefined minimum amount of time (at least twice the maximum propagation delay) before being able to bid for the floor. This is similar to the schemes for the provision of priority acknowledgments proposed for CSMA and ALOHA by Kleinrock and Tobagi [19].

To ensure that floor acquisition is enforced among competing senders hidden from one another and who have requested the floor (i.e., sent an RTS), the CTS sent by a receiver is guaranteed to last long enough (or to be repeated enough times) to jam any hidden sender that did not hear the RTS being acknowledged. This corresponds to a single-channel BTMA scheme in which sensing of error-free CTSs (for packet sensing) or the carrier of a CTS (for carrier sensing) over the data channel is used instead of the busy-tone signal.

When a station with data to send fails to acquire the floor or detects the floor being held by another station, it must reschedule its bid for the floor. This can be done using different persistence and backoff strategies. We also specify FAMA-NCS as using a uniform distribution when choosing backoff times; however, other backoff strategies can be adopted (e.g., see those proposed for MACAW [1]).

To simplify our analysis and description of FAMA-NCS, we do not address the effect of acknowledgments in the rest of this paper, and assume the simplest three-way handshake (RTS-CTS-data) with no acknowledgments.

B. FAMA-NCS

The length of a CTS in FAMA-NCS is larger than the aggregate of the length of an RTS plus one maximum round trip time across the channel, the transmit to receive turn around time, and any processing time. The length of an RTS is larger than the maximum channel propagation delay plus the transmit-to-receive turn around time and any processing time. This is required to avoid one station hearing a complete RTS before another has started to receive it. The relationship of the size of the CTS to the RTS gives the CTS *dominance* over the RTS in the channel. Once a station has begun transmission of a CTS, any other station within range of it that transmits an RTS simultaneously (i.e., within one propagation delay of the beginning of the CTS) will hear at least a portion of the dominating CTS and backoff, thereby letting the data packet that will follow to arrive free from collision. The dominating CTS plays the role of a busy tone.

Figure 1 specifies FAMA-NCS in detail. A station that has just been initialized must wait the time it takes to transmit the maximum-size data packet in the channel plus one maximum round-trip time across the channel. This allows any neighboring station involved in the process of receiving a data packet to complete the reception un-obstructed. The initialization time also gives the station the ability to learn of any local traffic in progress. If no carrier is detected during the initialization period, the station transitions to the PASSIVE state. Otherwise, it transitions to the REMOTE state. A station can only be in the PASSIVE state if it is properly initialized (i.e., has no packet to send, and senses an idle channel). In all other states, the station must have listened to the

channel for a time period that is sufficiently long for any neighbor involved in receiving data to have finished.

A station that is in the PASSIVE state and senses carrier transitions to the REMOTE state. On the other hand, a station that receives a packet to send in the PASSIVE state transmits an RTS and transitions to the RTS state. The sending station waits long enough for the destination to send the CTS. If the CTS packet is not received within the time allowed, the sender transitions to the BACKOFF state. If the sender hears noise on the channel after its RTS, it assumes a collision with a neighbor's dominating CTS and waits long enough for a maximum-length data packet to be received. Otherwise, upon receiving the CTS, the sender transmits its data packet. Because the CTS could be corrupted at the sender, once the destination station sends its CTS, it only needs to wait one maximum round-trip time to sense the beginning of the data packet from the source. If the data packet does not begin, the destination transitions either to the BACKOFF state (if it has traffic pending) or to the PASSIVE state.

In the BACKOFF state, if no carrier is detected during the entire backoff waiting period computed by the station, the station transmits an RTS and transitions to the RTS state as before. Otherwise, upon sensing carrier the station transitions to the REMOTE state.

For stations in the REMOTE state, FAMA-NCS enforces different waiting periods on passive stations (those stations not directly involved in the current transmission period) based on what was last heard on the channel. Any passive station that detects carrier transitions to the REMOTE state, and after the channel clears the waiting period is determined as follows:

- After hearing an RTS for another station, the station must wait long enough for a CTS to be transmitted by the destination and received by the sender, and the data packet to begin.
- After hearing a CTS from another station, the station must wait long enough to allow the other station to receive its data packet.
- After hearing a data packet, the waiting time is the enforced FAMA waiting period.
- After hearing noise (colliding control packets) on the channel, the waiting period must be long enough to allow another station time to receive a maximum size data packet.

The channel becomes idle when all stations are in either the PASSIVE or BACKOFF state. The next access to the channel is driven by the arrival of new packets to the network and retransmission of packets that have been backed off.

To increase the efficiency of the channel, a station that has successfully acquired the floor can dynamically send multiple packets together in a train, bounded by an upper limit. To allow this to be successful in a hidden-terminal environment, the destination station must alert its neighbors that it has more data packets coming, and to continue to defer their transmissions. FAMA-NCS uses a simple handshake mechanism to support packet trains.

If the sending station has multiple packets to send, it sets a MORE flag in the header of the data packet. When the destination receives the data packet and sees the MORE flag set, it immediately responds with a CTS, just as when hearing an RTS. This CTS alerts all neighbors that might interfere with the next data packet that they must continue to defer.

Additionally, stations in the REMOTE state must extend their waiting period after hearing a data packet with the MORE flag set to allow additional time for the sender to receive the CTS from the destination signaling that it can receive the next data packet.

III. COMPARATIVE THROUGHPUT ANALYSIS

We present an approximate throughput analysis that assumes the same traffic model first introduced for CSMA [9] to analyze the throughput of CSMA protocols. We compare the average throughput of FAMA-NCS against the throughput of non-persistent ALOHA and CSMA. The throughput of non-persistent CSMA used in this analysis was reported by Kleinrock and Tobagi [9].

```

Variable Definitions
CD = Carrier Detected
 $T_{PROC}$  = Maximum channel propagation delay
 $T_{PROCT}$  = Processing time for carrier detection
 $T_{TR}$  = Transmit to receive turn-around time
 $T_{WAIT} = 2 \times T_{PROP} + T_{PROC} + T_{TR}$ 
 $\gamma'$  = Time to transmit a RTS packet
 $\delta$  = Time to transmit a maximum sized data packet
Burst = Number of packets to send in a burst

Procedure START()
Begin
    Timer  $\leftarrow \delta + 2 \times T_{PROP}$ 
    While( $\overline{CD} \wedge$  Timer not expired) wait
    If (CD) Then call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
    Else call PASSIVE()
End

Procedure PASSIVE()
Begin
    While( $\overline{CD} \wedge$  No Local Packet) wait
    If (CD) Then call REMOTE(( $\delta + T_{WAIT}$ ),FALSE)
    Else Begin
        Burst  $\leftarrow$  maximum burst
        Transmit RTS Packet
        call RTS( $T_{WAIT}$ )
    End
End

Procedure RTS( $T_\sigma$ )
Begin
    Timer  $\leftarrow T_\sigma$ 
    While( $\overline{CD} \wedge$  Timer not expired) wait
    If (Timer Expired) Then call BACKOFF()
    Else Begin
        Receive Packet
        DO CASE of (received packet type)
        Begin
            CTS: call XMIT()
            Default: call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
        End
    End
End

Procedure XMIT()
Begin
    Wait  $T_{TR}$ 
    If ((Burst > 1)  $\wedge$  Local Packet)
        Then Begin
            Mark MORE flag in header
            Transmit Data Packet
            Burst  $\leftarrow$  Burst - 1
            call RTS( $T_{WAIT}$ )
        End
    Else Begin
        Transmit Data Packet
        Timer  $\leftarrow T_{WAIT}$ 
        While(Timer not expired) wait
        If (Local Packet) Then call BACKOFF()
        Else call PASSIVE()
    End
End

Procedure BACKOFF()
Begin
    Timer  $\leftarrow$  RANDOM(1,10  $\times$   $\gamma'$ )
    While( $\overline{CD} \wedge$  Timer not expired) wait
    If (CD) Then call REMOTE(( $\delta + T_{WAIT}$ ),FALSE)
    Else Begin
        Burst  $\leftarrow$  maximum burst
        Transmit RTS Packet
        call RTS( $T_{WAIT}$ )
    End
End

Procedure REMOTE( $T_\sigma, dfлаг$ )
Begin
    Timer  $\leftarrow T_\sigma$ 
    While( $\overline{CD} \wedge$  Timer not expired) wait
    If (Timer Expired)
        Then Begin
            If (Local Packet) Then call BACKOFF()
            Else call PASSIVE()
        End
    Else Begin
        Receive Packet
        DO CASE of (received packet type)
        Begin
            RTS: If ( $dfлаг$  = TRUE) call REMOTE( $T_\sigma$ ,TRUE)
            If(Destination ID = Local ID)
                Then Begin
                    Wait  $T_{TR}$ 
                    Transmit CTS Packet
                    call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
                End
            else: call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
            DATA: If(Destination ID = Local ID)
                Then Begin
                    Pass packet to upper layer
                    If (MORE flag set in header)
                        Then Begin
                            Transmit CTS
                            call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
                        End
                    Else Begin
                        If (MORE flag set in header)
                            Then Begin
                                call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
                            End
                        Else Begin
                            call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
                        End
                    End
                End
            END: call REMOTE(( $\delta + T_{WAIT}$ ),TRUE)
        End
    End
End

```

Fig. 1. FAMA-NCS Specification

We assume that there is an infinite number of stations who constitute a Poisson source sending RTS packets (for the case of FAMA), or new or retransmitted data packets (for the case of CSMA) to the channel with an aggregate mean generation rate of λ packets per unit time. Any station can listen to the transmissions of any other station. The transmission time of an RTS is $\gamma < \infty$, the transmission time of a CTS is $\gamma' < \infty$, the maximum transmission time of a data packet is $\delta < \infty$, and the hardware transmit-to-receive transition time is $\varepsilon < \infty$. There is no capture or fading on the channel. Any overlap by transmissions at a particular receiver causes that receiver to not understand either packet.

Each station is assumed to have at most one data block to send at any time. In all protocols, a station transmits the entire data block as a single packet (which is the case of CSMA and MACA as it is described in [8]) or as multiple packets (which is the case of FAMA-NCS). The average size of a data block is δ seconds. RTSs last γ seconds, CTSs last γ' seconds, and the maximum end-to-end propagation delay of the channel is τ seconds. Collisions (e.g., RTS packets in FAMA-NCS, data packets in CSMA) can occur in the channel, and we assume that, when a station has to retransmit a packet, it does so after a random retransmission delay that is much larger than δ on the average. The average channel utilization is given by [9]

$$S = \overline{U}/(\overline{B} + \overline{I}) \quad (1)$$

where \overline{B} is the expected duration of a busy period, defined to be a period of time during which the channel is being utilized; \overline{I} is the expected duration of an idle period, defined as the time interval between two consecutive busy periods; and \overline{U} is the time during a busy period that the channel is used for transmitting user data successfully.

The channel is assumed to introduce no errors, so packet collisions are the only source of errors, and stations detect such collisions perfectly. To further simplify the problem, we assume that two or more transmissions that overlap in time in the channel must all be retransmitted, and that a packet propagates to all stations in exactly τ seconds [9]. The later assumption provides a lower bound on the performance of the protocols we analyze.

Of course, this model is only a rough approximation of the real

case, in which a finite number of stations access the same channel, stations can queue multiple packets for transmission, and the stations' transmissions and retransmissions (of RTS or data packets) are correlated (e.g., a failed RTS is followed by another RTS within a bounded time).

For the case of non-persistent CSMA, we assume [9] that a separate perfect channel is used for acknowledgments to let a station know when its packet was received free of collisions, and that all acknowledgments are sent reliably. Therefore, the throughput of non-persistent CSMA used for comparison with FAMA protocols is only an upper bound.

To facilitate the comparison of the various protocols numerically, the graphs showing average throughput versus traffic load normalize the results obtained for S by making $\delta = 1$ and introducing the following variables:

$$a = \tau/\delta(\text{normalized propagation delay})$$

$$G = \lambda \times \delta(\text{offered Load, normalized to data packets})$$

For simplicity, we assume that any given node, w , has N neighbors. All nodes communicate over a single channel. The chosen node, w , is a Poisson source of RTSs with a mean rate of λ' floor requests (RTS packets). Additionally, each neighboring node is an independent Poisson source of RTSs as well, with a mean rate of λ' RTSs per second, such that the total floor requests seen in the channel at w equals $\lambda = \sum_1^{N+1} \lambda'$. The traffic generated at each node has its destination determined by a uniform distribution of the node's neighbors, i.e., for N neighbors, a node directs $1/N$ of its RTSs to each of those neighbors. This assumption makes our analysis independent of the specific routing choices made at each node. Again, each node is assumed to have at most one data block to send at any given time. A packet sent by any node propagates to all its neighbors (nodes in line of sight) in exactly τ seconds. All neighbor nodes are within line-of-sight and range of node w , but may be hidden from one another. The population of neighbors at node w is partitioned such that, for any given neighbor of w , there are Q neighbors that are hidden from it, and $N - 1 - Q$ neighbors that it can hear in addition to w . This connectivity assumption is a simplification of the fact that the neighbors of a node may form

multiple fully connected subgroups; it provides a lower bound on performance for the case in which $Q = N$ and results in a fully-connected network for the case in which $Q = 0$.

Assuming the above model, the following theorem gives the throughput for a given node.

Theorem 1: The throughput of FAMA-NCS for a system with N independent neighbors, of which Q are hidden from the others is given by

$$S = \frac{\delta \cdot e^{-\lambda' (N\tau + Q(2\gamma - \tau))}}{e^{-\lambda' (N\tau + Q(2\gamma - \tau))} [(3Q + N + 4)\tau + (Q - N)\gamma + (Q + 1)(\gamma' + \delta)]} \dots \\ \dots + (N + 1) \cdot [T_{F_2} + (\gamma + \tau - T_{F_2}) \cdot e^{-\lambda'^2 Q \gamma} + \frac{1}{\lambda'} + N \cdot (\gamma + \gamma' + \delta \cdot \tau)] \\ \text{Where } T_{F_2} \text{ equals: } \left[\frac{e^{\lambda_Q \gamma} - 1 - \lambda_Q \gamma}{\lambda_Q \gamma (1 - e^{-\lambda_Q \gamma})} \right] + \gamma \quad (2)$$

Proof:

An RTS originated from any node A is successful at w if no other RTS from any other node in w 's neighborhood collides with A 's RTS. At node w and the $N - 1 - Q$ neighbors that hear A the vulnerability period of A 's RTS is τ seconds because these nodes can detect carrier τ seconds after the beginning of the RTS. Accordingly, because of the independence of the nodes transmissions, an RTS is successful within the $N - 1 - Q$ nodes and w with probability

$$\prod_1^{N-Q} e^{-\lambda' \tau} \quad (3)$$

In contrast, the vulnerability period of an RTS with respect to the other Q hidden nodes is 2γ because these nodes cannot hear A 's transmissions. Accordingly, an RTS is successful with respect to any of the other Q nodes with probability $e^{-\lambda'(2\gamma)}$. Because all nodes are identical and independent, it follows that an RTS from a node A is successful at node w with the following probability:

$$P_S = \prod_1^{(N-Q)} e^{-\lambda' \tau} \prod_1^Q e^{-\lambda'(2\gamma)} = e^{-\lambda' (N\tau + Q(2\gamma - \tau))} \quad (4)$$

The probability that a given RTS arrival was generated by node w is $(\lambda'/\lambda) \equiv \frac{1}{N+1}$. The traffic directed to w comes from each of w 's N neighbors, who generate RTSs at a rate of λ' with $1/N$ destined for w . Accordingly, the probability that a given RTS arrival is meant for w is

$$\frac{\sum_{i=1}^N \left[\frac{1}{N} \cdot \lambda' \right]}{\lambda} = \lambda'/\lambda = \frac{1}{N+1} \quad (5)$$

A successful transmission period in the time line of node w includes the RTS with a one-way propagation delay, followed by the return CTS with a one-way propagation delay and lastly the data packet with a one-way propagation delay for a total of $T = \gamma + \gamma' + \delta + 3\tau$.

There are two types of failed transmission periods for RTSs. If only the nodes that can hear each other send RTSs in a transmission period, the average duration of the period in the time line of node w equals $T_{F_1} = \gamma + \bar{Y}$, where \bar{Y} is the same as in the fully-connected network case analyzed in [5]. Noting that $\bar{Y} \leq \tau$, given that $\tau \ll \gamma$, we use the upper bound $\gamma + \tau$ for T_{F_1} .

If one or more of the hidden nodes send RTSs in a failed transmission period, the failed transmission period consists of multiple overlapping RTSs with durations of γ seconds. Because these nodes are hidden and independent of one another, the length of the average failed transmission period in this case can be obtained by treating this case as an ALOHA channel with Q different nodes, in which a node A has a rate of λ' . To make use of prior results, we make the simplifying assumption that N and Q are very large.

The aggregate RTS arrivals for these nodes is $\lambda_Q = \sum_1^Q \lambda'$. Substituting λ_Q for λ and γ for δ in the expression of \bar{Y} derived in [16] we obtain

$$T_{F_2} = \left[\frac{e^{\lambda_Q \gamma} - 1 - \lambda_Q \gamma}{\lambda_Q \gamma (1 - e^{-\lambda_Q \gamma})} \right] + \gamma \quad (6)$$

The busy period can be partitioned into three categories – periods of successful transmissions at node w , periods of failed transmissions at node w , and periods when node w is deferred due to neighbors receiving data from nodes hidden from w .

At node w , successful transmission periods can consist of transmissions from w , transmissions to w , or successful transmission overheard by w . The probability that a given RTS is from w itself is λ'/λ , and the probability that it is successful is P_S . Therefore, the time allotted to successful transmission for w is $(\lambda'/\lambda) \cdot P_S \cdot T$.

The probability of a transmission directed to w from a given neighbor is $\frac{1}{N} \lambda'$ and with probability P_S such a transmission is successful. Therefore, the total time for successful transmissions to w from all of its neighbors is $(\lambda'/\lambda) \cdot P_S \cdot T$.

We say that a transmission is overheard by node w if it is sent by a neighbor of w , but is not meant for w . For any given neighbor, this is equal to the remaining number of transmissions not meant for w , i.e., $(\frac{N-1}{N}) \lambda'$. The total overheard transmissions from all neighbors is then $((N-1) \lambda') / \lambda$. As such, the total time for overheard successful transmissions is $[((N-1) \lambda') / \lambda] \cdot P_S \cdot T$. Adding the time spent in successful transmissions for w and overheard successful transmissions together, we obtain that the total time for successful transmissions at node w is $P_S \cdot T$.

The average failed period consists of failed transmissions involving the hidden nodes and those that involve only the nodes that can hear the source.

The probability that none of the hidden neighbors of node w are involved in a failed transmission at w is the probability that they have no arrivals during an RTS time, which is $e^{-\lambda'^2 Q \gamma}$. The probability that an RTS fails because of interference from one of the $(N-Q)$ fully connected nodes around w is $(1 - e^{-\lambda'(N-Q)\tau})$.

The probability of having a failed transmission period at node w when only connected neighbors of w interfere is the probability that no hidden neighbor transmits within the duration of an RTS and some of the connected neighbors transmit within τ seconds of an RTS. Because nodes send RTSs independently of one another, this probability equals $(e^{-\lambda'^2 Q \gamma} - e^{-\lambda'(N\tau + Q(2\gamma - \tau))})$.

Similarly, the probability of having a failed transmission period at node w involving only neighbor nodes hidden from one another is $(1 - e^{-\lambda'^2 Q \gamma})$. It then follows that the average failed transmission period lasts

$$T_{FAIL} = (e^{-\lambda'^2 Q \gamma} - e^{-\lambda'(N\tau + Q(2\gamma - \tau))}) \cdot T_{F_1} + (1 - e^{-\lambda'^2 Q \gamma}) \cdot T_{F_2} \quad (7)$$

We say that node w is deferred when one of its neighbors is receiving a data packet and after hearing noise on the channel (i.e., a failed period in which w was not directly involved). In both instances, w defers for a maximum packet transmission time, T . The probability that w hears noise from its neighbors' transmissions and defers is $((N \cdot \lambda')/\lambda) \cdot (1 - P_S)$. Each of w 's neighbors is identical to w and as such receives packets directed to it at a rate of λ' . Of that traffic sent by the nodes hidden from w , w can only hear the neighbor's CTS and will defer. As such, the probability that w is deferred by a neighbor receiving traffic not overheard by w is $((Q \cdot \lambda')/\lambda) \cdot P_S$. Therefore, the average time during which w is deferred by neighbor traffic (either noise or successful data) is

$$T_{DEFER} = [((N \cdot \lambda')/\lambda) \cdot (1 - P_S) + ((Q \cdot \lambda')/\lambda) \cdot P_S] \cdot T \\ = \left[\frac{N}{N+1} + \frac{Q-N}{N+1} \cdot P_S \right] \cdot T \quad (8)$$

Accordingly, the average busy period lasts

$$\bar{B} = P_S \cdot T + T_{FAIL} + T_{DEFER} \quad (9)$$

Substituting Eqs. (7) and (8) in the above Eq., we obtain

$$\begin{aligned} \bar{B} &= \left(e^{-\lambda' \cdot 2Q\gamma} - e^{-\lambda' (N\tau + Q(2\gamma - \tau))} \right) \cdot T_{F_1} + (1 - e^{-\lambda' \cdot 2Q\gamma}) \cdot T_{F_2} \\ &\quad + T \cdot \left[\frac{N}{N+1} + \frac{Q+1}{N+1} \cdot P_S \right] \end{aligned} \quad (10)$$

The average idle period lasts 2τ seconds after every successful data packet transmission plus an average interarrival time of RTSs from all groups; therefore, we have

$$\bar{I} = \frac{1}{\lambda} + 2\tau \cdot P_S \quad (11)$$

The average utilization time at node w is simply the proportion of time in which useful data are sent during a successful busy period in w 's proximity, and

$$\bar{U} = \frac{1}{(N+1)} \cdot \delta \cdot P_S \quad (12)$$

Substituting Eqs. (10), (11) and (12) into Eq. (1) we obtain Eq. (2). \square

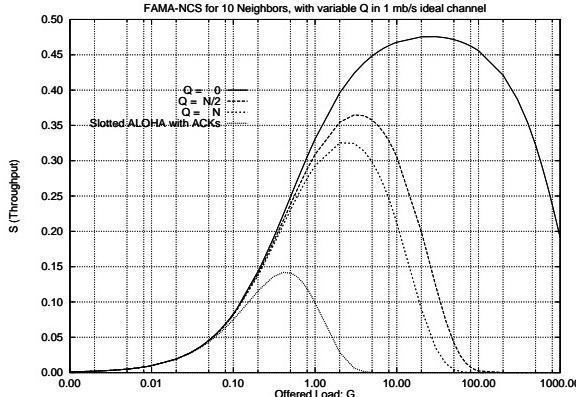


Fig. 2. Throughput versus load for various Q in FAMA-NCS network using *ideal* radio parameters

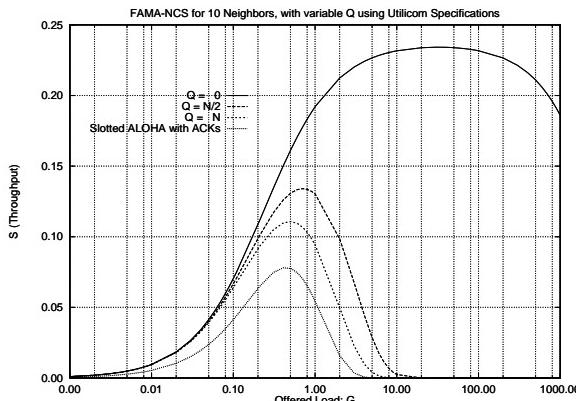


Fig. 3. Throughput versus load for various Q in FAMA-NCS network using *Utilicom* radio parameters

To compare the performance of FAMA-NCS in a multihop network, we use slotted ALOHA using non-priority acknowledgments. This choice is driven by the following considerations: (a)

with hidden terminals, CSMA degrades to pure ALOHA; (b) implementing ALOHA with a single channel requires the use of acknowledgments to let the senders' of packets know if they need to retransmit; (c) implementing priority acknowledgment schemes (e.g., the schemes proposed for ALOHA and CSMA by Tobagi and Kleinrock [19]) do not work well with hidden terminals; and (d) slotted ALOHA has better performance than pure ALOHA.

Figures 2 and 3 show the results of this comparison using the value of throughput derived in [19] for slotted ALOHA with acknowledgments. We assume a network where each node has 10 neighbors for varying values of Q : 0, where the network is fully connected; $N/2$, where half the neighbors of any node are hidden from a neighbors' neighbors; and $(N-1)$, where all the nodes are hidden from their neighbors' neighborhoods. The latter case corresponds to a hypercube topology. Additionally, we assume a network with a propagation delay of $6\mu s$ (one mile), 500-byte data packets, 25-byte RTS, 50-byte CTS. We show results for both a 1 Mb/s channel with zero preamble and processing overhead, and a 298 kb/s channel with processing and preamble based on the specifications of the Utilicom model 2020 radio transceiver. The Utilicom radio has a turn-around time of about 5ms to ramp up the transmitter (including the preamble), and about the same to ramp down and be ready to receive again. At 298 kb/s, the RTS itself is about 1 ms long; therefore, it takes about 11ms for a transmitter to send an RTS and be ready to receive a CTS. Together with its transmission overhead, a 500-byte data packet then becomes about 25ms in length. This makes the ratio of RTS to data about 0.47, which severely degrades the performance of FAMA protocols [5].

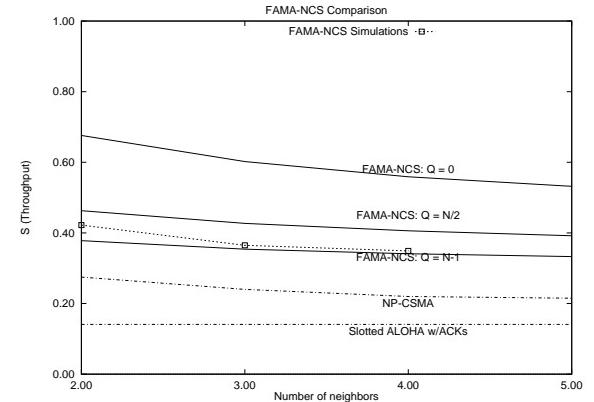


Fig. 4. Throughput versus degree of node in ad-hoc network

Figure 4 shows the maximum throughput for FAMA-NCS in multihop networks versus the number of neighbors per node for the values of $Q(0, \frac{N}{2}, N-1)$. The analysis assumes a 1Mb/s wireless network with a diameter of 1 mile (propagation delay of about $6\mu s$). The size of data packets RTSs used were 500 and 25 bytes, respectively. The figure also shows the throughput for slotted ALOHA with acknowledgments, which reflects the expected behavior of both ALOHA and CSMA protocols operating over a single channel when hidden terminals abound. We include the performance of non-persistent CSMA predicted by the model developed by Kleinrock and Tobagi when hidden terminals exist [17]; this value of throughput is only an upper bound on CSMA, because it assumes a separate ideal channel over which acknowledgments to packets are sent correctly in zero time.

The above results clearly show the importance of floor acquisition, which makes FAMA-NCS a far better choice than CSMA for a multihop setting. The results also indicate the important role that radio parameters play in the overall performance of an ad-hoc network. Without good carrier sensing and turn-around times, throughput degrades substantially.

IV. SIMULATION RESULTS

To validate the approximations made in our performance analysis of FAMA-NCS, we carried out a number of simulations*. The simulations ran the actual code used to implement the MAC protocols in embedded systems and, for the case of FAMA-NCS, this code is based on the specifications shown in Figure 1.

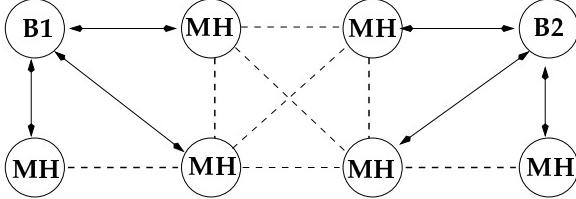


Fig. 5. Simulation topology used in second set of experiments

In the first set of experiments, a 1Mb/s wireless network was modeled with stations at one mile from neighbors (propagation delay of approximately $6\mu s$). Data packet size was 500 bytes, and RTS and CTS were 25 and 48 bytes respectively. As Figure 4 illustrates, the simulation results are almost identical to the analytical results for the case in which $Q = (N - 1)$ (all neighbors hidden from each other), which validates the approximations used to make our model tractable.

In the second set of experiments, we assumed a 1 Mbps network with the same topology of Configuration shown in Figure 5. Traffic was only between $B1$ and its neighbors, and between $B2$ and its neighbors. Table IV lists the results for FAMA-NCS, IEEE 802.11 DFWMAC, and MACAW. In the table, “total input” refers to traffic correctly received and meant for any node; “local input” refers to traffic correctly received and meant for the receiving node. The results illustrate that making the CTSs dominate the RTSs, i.e., enforcing floor acquisition, is important for throughput in ad-hoc networks.

Avg. Rate Pkts. Received	FAMA-NCS (2KB pkts)	IEEE 802.11 (2KB pkts)	MACAW (1KB pkts)
Avg. Total Input	36.0KB/s	17.0KB/s	—
Avg. Local Input	15.3KB/s	8.4KB/s	1.1KB/s
Avg. at N1 & N4	15.5KB/s	5.5KB/s	2.1KB/s
Avg. for others	15.2KB/s	9.3KB/s	0.8KB/s

Fig. 6. Throughput of FAMA-NCS, MACAW and IEEE 802.11

V. CONCLUDING REMARKS

FAMA-NCS permits a sender to acquire control of the channel in the vicinity of a receiver dynamically before transmitting data packets. The floor acquisition strategy uses an RTS-CTS handshake and is based on a few simple principles: (a) making the senders listen to the channel before transmitting RTSs; (b) implementing a busy-tone mechanism using a single channel and half-duplex radios by making the receiver send CTSs that last long enough for the hidden senders to realize that they must back off; and (c) providing priority to those stations who successfully complete a handshake.

Although many MAC protocols have been introduced in the past based on RTS-CTS exchanges, FAMA-NCS is the first single-channel protocol to provide the equivalent functionality of a busy-tone solution. Contrary to the conjectures made in prior work on MAC protocols based on collision avoidance [1], [8], our analysis demonstrates that carrier sensing should be used in single channel networks because it substantially improves performance by enabling floor acquisition in the presence of hidden terminals.

FAMA-NCS has been successfully implemented and demonstrated in actual packet radios for ad-hoc networks [21] built using commercial direct-sequence spread-spectrum radios and controllers.

REFERENCES

- [1] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, “MACAW: A Media Access Protocol for Wireless LAN’s,” in *Proc. ACM SIGCOMM ’94*, pp. 212–25, London, UK, Aug. 31–Sept. 2, 1994 1994.
- [2] K. Biba, “A Hybrid Wireless MAC Protocol Supporting Asynchronous and Synchronous MSDU Delivery Services,” Tech. Rep. Paper 802.11/91-92, IEEE 802.11 Working Group, 1992.
- [3] K.-C. Chen, “Medium access control of wireless lans for mobile computing,” *IEEE Network*, vol. 8, no. 5, pp. 50–63, 1994.
- [4] A. Colvin, “CSMA with collision avoidance,” *Computer Commun.*, vol. 6, no. 5, pp. 227–35, 1983.
- [5] C. L. Fullmer and J.J. Garcia-Luna-Aceves, “Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks,” *Proc. ACM SIGCOMM 95*, Cambridge, MA, Aug. 28–Sept. 1, 1995.
- [6] C. L. Fullmer and J.J. Garcia-Luna-Aceves, “Solutions to Hidden Terminal Problems in Wireless Networks,” *Proc. ACM SIGCOMM 97*, Cannes, France, Sep. 14–18, 1997.
- [7] *Wireless LAN Medium Access Control (MAC) and Physical Specifications, IEEE std. 802.11*, IEEE, June 1997.
- [8] P. Karn, “MACA - a new channel access method for packet radio,” in *ARRL/CRRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.
- [9] L. Kleinrock and F. A. Tobagi, “Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics,” *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.
- [10] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, eds., *Proceedings of the IEEE*, vol. 75, IEEE, January 1987.
- [11] W. F. Lo and H. T. Mouftah, “Carrier Sense Multiple Access with Collision Detection for Radio Channels,” in *IEEE 13th Int’l Commun. and Energy Conf.*, pp. 244–47, IEEE, 1984.
- [12] R. M. Metcalfe and D. R. Boggs, “ETHERNET: Distributed packet switching for local computer networks,” *Communications of the ACM*, vol. 19, no. 7, pp. 395 – 403, 1976.
- [13] R. Rom, “Collision Detection in Radio Channels,” in *Local area and multiple access networks*, pp. 235–49, Computer Science Press, 1986.
- [14] R. Rom and M. Sidi, *Multiple Access Protocols Performance and Analysis*. Springer-Verlag, 1990.
- [15] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.
- [16] H. Takagi and L. Kleinrock, “Output processes in contention packet broadcasting systems,” *IEEE Trans. Commun.*, vol. COM-33, no. 11, pp. 1191–9, 1985.
- [17] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels: Part II - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution,” *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1417–1433, 1975.
- [18] F. A. Tobagi and L. Kleinrock, “Packet switching in radio channels: Part III - polling and (dynamic) split-channel reservation multiple access,” *IEEE Trans. Commun.*, vol. COM-24, no. 8, pp. 832–845, 1976.
- [19] F. A. Tobagi and L. Kleinrock, “The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels,” *IEEE Trans. Commun.*, vol. COM-26, no. 6, pp. 815–826, 1978.
- [20] B. Vaduvu, *Access, Addressing and Security in Wireless Packet Networks*. PhD thesis, University of California, Berkeley, Computer Science Department, 1995.
- [21] *WINGS for The Internet*, <http://www.cse.ucsc.edu/research/ccrg/projects/wings.html>.
- [22] C. Wu and V. O. K. Li, “Receiver-initiated busy-tone multiple access in packet radio networks,” *ACM SIGCOMM 87 Workshop: Frontiers in Computer Communications Technology*, Stowe, VT, USA, 11–13 Aug. 1987.

* We thank Rooftop Communications Corporation for donating the *C⁺⁺* Protocol Toolkit (CPT) simulator.